Local Processing Prototyping: Benchmark

Juan Miguel Carceller and Pip Hamilton

University College London, Imperial College London

February 22, 2021

Outline

- Last call, answering the question of How much processing can we done for monitoring came up
- Limits were set on 1 core per APA
- We took some raw data from ProtoDUNE and implemented a simple benchmark

The data

- We are using data coming from long-readouts
- `https://wiki.dunescience.org/wiki/ProtoDUNE-SP_long_TPC_readout`
- We downloaded to our pcs the file `felix-2020-06-02-101141.0.2.0.bin` (how to make cmake work in dunegpvms?)
- Around 1 GB, 1 s of data
- The file contains arrays of WIB frames, data coming from a single FELIX link (256 channels)
- Frames are separated by 25 ticks of the master clock (2 MHz) = 500 ns

The benchmark

- We use a modified version of a script by Phil. Rodrigues to read data from binary:
- dumpfile-to-text.cpp would dump all the info to a text file
- `https://github.com/philiprodrigues/felix-long-readout-tools`
- Modifications: it's the input to our light histogram class, it's also timed and it doesn't output all the info to a text file as before
- `CMakeLists.txt` modified to include optimization flags

Histogram class

- To avoid using ROOT, a very simple histogram class was implemented with the basic functions of TH1

hist.h

```
#include <vector>
#include <string>

class Hist {

  int FindBin(double x);

public:
  double fLow, fHigh, fStepSize;
  int fNentries;
  double fSum;

  int fSteps;
  std::vector<int> fEntries;

  Hist(int steps, double low, double
      high);
  int Fill(double x);

  void Save(std::string filename);
  void Save(std::ofstream &filehandle);
};
```

Histogram class

- To avoid using ROOT, a very simple histogram class was implemented with the basic functions of TH1

hist.cpp

```
Hist::Hist(int steps, double low,
    double high)
  : fSteps(steps), fLow(low), fHigh(
      high)
{
  fEntries = std::vector<int> (steps,
      0);
  fStepSize = (high - low) / steps;
}
```

Histogram class

- To avoid using ROOT, a very simple histogram class was implemented with the basic functions of TH1

hist.cpp

```
int Hist::FindBin(double x){
  return (x - fLow) / fStepSize;
}

int Hist::Fill(double x){
  int bin = FindBin(x);
  // Underflow, do nothing
  if(bin < 0) return -1;

  // Overflow, do nothing
  if(bin >= fSteps) return -1;

  fEntries[bin]++;
  fNentries++;
  fSum += x;
  return bin;
}
```

Histogram class

- To avoid using ROOT, a very simple histogram class was implemented with the basic functions of TH1

hist.cpp

```
void Hist::Save(std::ofstream &
    filehandle){
  filehandle << fSteps << " " << fLow
     << " " << fHigh << " " << std::
     endl;
  for (auto x: fEntries)
    filehandle << x << " ";
  filehandle << std::endl;
}
```

Results

- The benchmark is run for batches of 10k frames (5 ms of data-taking, more or less similar to a DUNE event)
- For each batch a histogram is filled for every channel and saved to a text file

---

Output:

Total elapsed (sec, wall time): 0.373
Total elapsed (sec, processing batch time without saving): 0.0448734
Total elapsed (sec, processing batch time with saving): 0.369829

Some math

- Only processing, 0.045 s for 214 batches of 10k frames (5 ms), that's .2 ms for each batch for a single histogram per channel - 2 ms per APA
- Counting saving time to a text file (may not be optimal), it's 1.73 ms - 17.3 ms per APA
- Adding another histogram increases time by ~ 2

Cross-check

- A small pyROOT script is used to check that these histograms that we are saving are good

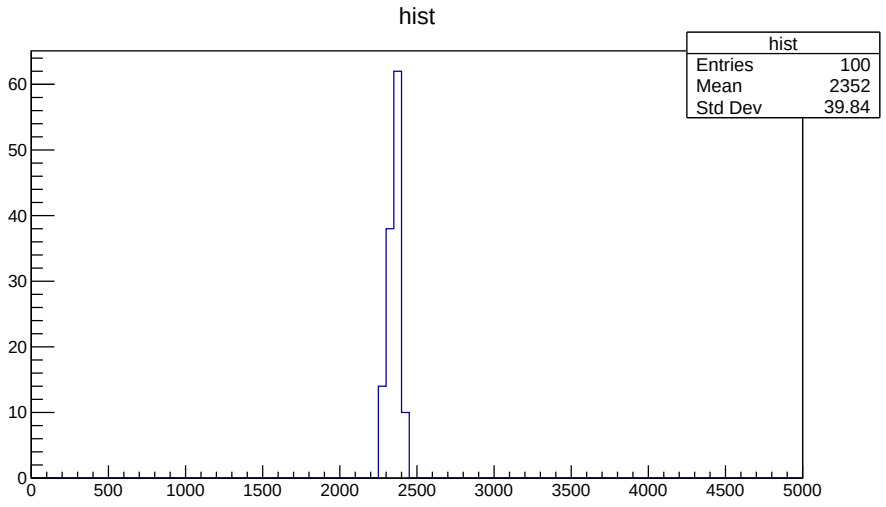
```
import ROOT

f = open('hist_12.txt')
steps, low, high = list(map(int, f.readline().split()))
entries = list(map(int, f.readline().split()))

h = ROOT.TH1F('hist', 'hist', steps, low, high)
for i in range(1, len(entries)+1):
    h.SetBinContent(i, entries[i-1])
h.Draw()
input('Press enter to finish')
```

Output

- One example of histogram from the text file outputs



Output

- One file for each batch

```
c1.pdf                  hist_10.txt  hist_26.txt  hist_42.txt  hist_58.txt  hist_74.txt  hist_90.txt  hist_106.txt  hist_122.txt  hist_138.txt  hist_154.txt  hist_170.txt  hist_186.txt  hist_202.txt
cmake_install.cmake     hist_11.txt  hist_27.txt  hist_43.txt  hist_59.txt  hist_75.txt  hist_91.txt  hist_107.txt  hist_123.txt  hist_139.txt  hist_155.txt  hist_171.txt  hist_187.txt  hist_203.txt
CMakeCache.txt          hist_12.txt  hist_28.txt  hist_44.txt  hist_60.txt  hist_76.txt  hist_92.txt  hist_108.txt  hist_124.txt  hist_140.txt  hist_156.txt  hist_172.txt  hist_188.txt  hist_204.txt
CMakeFiles              hist_13.txt  hist_29.txt  hist_45.txt  hist_61.txt  hist_77.txt  hist_93.txt  hist_109.txt  hist_125.txt  hist_141.txt  hist_157.txt  hist_173.txt  hist_189.txt  hist_205.txt
custom-dumpfile-to-text hist_14.txt  hist_30.txt  hist_46.txt  hist_62.txt  hist_78.txt  hist_94.txt  hist_110.txt  hist_126.txt  hist_142.txt  hist_158.txt  hist_174.txt  hist_190.txt  hist_206.txt
hist.txt                hist_15.txt  hist_31.txt  hist_47.txt  hist_63.txt  hist_79.txt  hist_95.txt  hist_111.txt  hist_127.txt  hist_143.txt  hist_159.txt  hist_175.txt  hist_191.txt  hist_207.txt
hist_0.txt              hist_16.txt  hist_32.txt  hist_48.txt  hist_64.txt  hist_80.txt  hist_96.txt  hist_112.txt  hist_128.txt  hist_144.txt  hist_160.txt  hist_176.txt  hist_192.txt  hist_208.txt
hist_1.txt              hist_17.txt  hist_33.txt  hist_49.txt  hist_65.txt  hist_81.txt  hist_97.txt  hist_113.txt  hist_129.txt  hist_145.txt  hist_161.txt  hist_177.txt  hist_193.txt  hist_209.txt
hist_2.txt              hist_18.txt  hist_34.txt  hist_50.txt  hist_66.txt  hist_82.txt  hist_98.txt  hist_114.txt  hist_130.txt  hist_146.txt  hist_162.txt  hist_178.txt  hist_194.txt  hist_210.txt
hist_3.txt              hist_19.txt  hist_35.txt  hist_51.txt  hist_67.txt  hist_83.txt  hist_99.txt  hist_115.txt  hist_131.txt  hist_147.txt  hist_163.txt  hist_179.txt  hist_195.txt  hist_211.txt
hist_4.txt              hist_20.txt  hist_36.txt  hist_52.txt  hist_68.txt  hist_84.txt  hist_100.txt  hist_116.txt  hist_132.txt  hist_148.txt  hist_164.txt  hist_180.txt  hist_196.txt  hist_212.txt
hist_5.txt              hist_21.txt  hist_37.txt  hist_53.txt  hist_69.txt  hist_85.txt  hist_101.txt  hist_117.txt  hist_133.txt  hist_149.txt  hist_165.txt  hist_181.txt  hist_197.txt  hist_213.txt
hist_6.txt              hist_22.txt  hist_38.txt  hist_54.txt  hist_70.txt  hist_86.txt  hist_102.txt  hist_118.txt  hist_134.txt  hist_150.txt  hist_166.txt  hist_182.txt  hist_198.txt  Makefile
hist_7.txt              hist_23.txt  hist_39.txt  hist_55.txt  hist_71.txt  hist_87.txt  hist_103.txt  hist_119.txt  hist_135.txt  hist_151.txt  hist_167.txt  hist_183.txt  hist_199.txt  read.py
hist_8.txt              hist_24.txt  hist_40.txt  hist_56.txt  hist_72.txt  hist_88.txt  hist_104.txt  hist_120.txt  hist_136.txt  hist_152.txt  hist_168.txt  hist_184.txt  hist_200.txt
hist_9.txt              hist_25.txt  hist_41.txt  hist_57.txt  hist_73.txt  hist_89.txt  hist_105.txt  hist_121.txt  hist_137.txt  hist_153.txt  hist_169.txt  hist_185.txt  hist_201.txt
```

Output

- Example of a file, two rows for each histogram, the first one with the number of steps, low limit and high limit and the second row with the entries in each bin

```
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 0 5000
```

Backup